Amendments to the Claims

Please amend Claims 1, 9, 11, 17, 19, 23, 25 and 29. The Claim Listing below will replace all prior versions of the claims in the application:

Claim Listing

1.   (Currently amended) A method of analyzing multi-threaded programs, comprising:

determining that a resource of interest is susceptible to unsynchronized accesses to a resource of interest can be performed by a plurality of threads, such that there can be latent unsynchronized accesses to the resource;

causing occurrence of a latent unshnchronized access to the resource by:

(a)   receiving a request from a first thread to access the resource;

(b)   suspending the first thread for requesting to access the resource to which unsynchronized accesses can be performed; and

(c)   while the first thread is suspended, receiving a request from a second thread to access the resource, and

from said caused latent unsynchronized access, detecting unsynchronized accesses to the resource.

2.   (Original) The method of claim 1, wherein the requests of the first and second threads is to write data to the resource.

3.   (Original) The method of claim 1, further comprising awakening the first thread.

4.   (Original) The method of claim 1, further comprising logging for a user that the first and second thread performed unsynchronized accesses to the resource.

5.   (Original) The method of claim 1, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

6.   (Previously presented) The method of claim 5, wherein the first thread is also suspended on an event, meaning that the event awakens the first thread.

7.   (Original) The method of claim 6, wherein the second thread sends the event that awakens the first thread.

8.   (Original) The method of claim 1, wherein the resource is a memory location, region of memory, hardware component, or peripheral device.

9.   (Currently amended) A computer program product for analyzing multi-threaded programs, comprising:

        computer code that determines that ~~unsynchronized accesses to~~ a resource of interest ~~can be performed~~ is susceptible to unsynchronized accesses by a plurality of threads, such that there can be latent unsynchronized accesses to the resource;

        computer code that causes occurrence and detection of a latent unsynchronized access to the resource by:

                (a) ~~receives~~ receiving a request from a first thread to access the resource;

                (b) ~~computer code that suspends~~ suspending the first thread for requesting to access the resource ~~to which unsynchronized accesses can be performed~~; and

                (c) ~~computer code that~~ while the first thread is suspended, ~~receives~~ receiving a request from a second thread to access the resource; and

        a computer readable medium that stores the computer codes.

10.   (Original) The computer program product of claim 9, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

11.   (Currently amended) A method of analyzing multi-threaded programs, comprising:

determining ~~that~~ susceptibility of a memory location to unsynchronized accesses ~~to a memory location can be performed~~ by a plurality of threads, such that there can be latent unsynchronized accesses to the memory location;

causing occurrence of a latent unsynchronized access to the memory location and detecting said latent unsynchronized access by:

(a) receiving a request from a first thread to write data to the memory location;

(b) suspending the first thread for requesting to write data to the memory location ~~to which unsynchronized accesses can be performed~~; and

(c) while the first thread is suspended, receiving a request from a second thread to write data to the memory location.

12.   (Original) The method of claim 11, further comprising awakening the first thread.

13.   (Original) The method of claim 11, further comprising logging for a user that the first and second thread performed unsynchronized writes to the memory location.

14.   (Original) The method of claim 11, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

15.   (Previously presented) The method of claim 14, wherein the first thread is also suspended on an event, meaning that the event awakens the first thread.

16.   (Original) The method of claim 15, wherein the second thread sends the event that awakens the first thread.

17.   (Currently amended) A computer program product for analyzing multi-threaded programs, comprising:

computer code that determines ~~that~~ susceptibility of a memory location to unsynchronized accesses ~~to a memory location can be performed~~ by a plurality of threads, such that there can be latent unsynchronized accesses to the memory location;

computer code that causes and detects occurrence of a latent unsynchronized access to the memory location by:

    (a) ~~receives~~ receiving a request from a first thread to write data to the memory location;

    (b) ~~computer code that suspends~~ suspending the first thread for requesting to write data to the memory location ~~to which unsynchronized accesses can be performed~~; and

    (c) ~~computer code that~~ while the first thread is suspended, ~~receives~~ receiving a request from a second thread to write data to the memory location; and

a computer readable medium that stores the computer codes.

18.   (Original) The computer program product of claim 17, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

19.   (Currently amended) A method of analyzing multi-threaded programs, comprising:

    determining that latent unsynchronized accesses to a memory location can be performed by a plurality of threads;

    causing occurrence of a latent unsynchronized access to the memory location by:

    (a) receiving a request from a first thread to write data to the memory location;

    (b) suspending the first thread for requesting to write data to the memory location ~~to which unsynchronized accesses can be performed~~; and

    (c) while the first thread is suspended, receiving a request from a second thread to write data to the memory location; and

    (d) awakening the first thread; and

logging for a user that the first and second thread performed unsynchronized writes to the memory location.

20. (Original) The method of claim 19, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

21. (Previously presented) The method of claim 20, wherein the first thread is also suspended on an event, meaning that the event awakens the first thread.

22. (Original) The method of claim 21, wherein the second thread sends the event that awakens the first thread.

23. (Currently amended) A computer program product for analyzing multi-threaded programs, comprising:

computer code that determines that <u>latent</u> unsynchronized accesses to a memory location can be performed by a plurality of threads;

computer code that <u>causes occurrence of a latent unsynchronized access to the memory location by:</u>

(a) ~~receives~~ <u>receiving</u> a request from a first thread to write data to the memory location;

(b) ~~computer code that suspends~~ <u>suspending</u> the first thread for requesting to write data to the memory location ~~to which unsynchronized accesses can be performed~~;

(c) ~~computer code that~~ while the first thread is suspended, ~~receives~~ <u>receiving</u> a request from a second thread to write data to the memory location; <u>and</u>

(d) ~~computer code that awakens~~ <u>awakening</u> the first thread; <u>and</u>

computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location; and

a computer readable medium that stores the computer codes.

24. (Original) The computer program product of claim 23, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

25. (Currently amended) A method of analyzing multi-threaded programs, comprising:

modifying an existing multi-threaded program <u>to</u> include computer code that determines that <u>latent</u> unsynchronized accesses to a memory location can be performed by a plurality of threads;

modifying the existing multi-threaded program to include computer code that <u>causes occurrence of a latent unsynchronized access to the memory location by</u> <u>suspending</u> ~~suspends~~ a first thread for requesting to write data to [[a]] <u>the</u> memory location ~~to which unsynchronized accesses can be performed~~, wherein a second thread writes data to the memory location; and

modifying the existing multi-threaded program to include computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location when a second thread writes data to the memory location.

26. (Original) The method of claim 25, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

27. (Previously presented) The method of claim 26, wherein the first thread is also suspended on an event, meaning that the event awakens the first thread.

28. (Original) The method of claim 27, wherein the second thread sends the event that awakens the first thread.

29. (Currently amended) A computer program product for analyzing multi-threaded programs, comprising:

computer code that modifies an existing multi-threaded program to include computer code that determines that latent unsynchronized accesses to a memory location can be performed by a plurality of threads;

computer code that modifies the existing multi-threaded program to include computer code that causes occurrence of a latent unsynchronized access to the memory location by suspending ~~suspends~~ a first thread for requesting to write data to [[a]] the memory location ~~to which unsynchronized accesses can be performed~~, wherein a second thread writes data to the memory location;

computer code that modifies the existing multi-threaded program to include computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location when a second thread writes data to the memory location; and

a computer readable medium that stores the computer code.

30.   (Original) The computer program product of claim 29, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.